## [html](#目录)

### 简要

<http://www.w3school.com.cn/>

sublime text 3使用，输入!,ctrl+e；

<https://developer.mozilla.org/zh-CN/docs/Web/HTML>

### 发布自己的网站

浏览器通过请求从web服务器读取页面，类似：http://www.... 如何显示是浏览器的事；

所有网页的集合叫网站，要发布才能让别人看到（把网站放到一个web服务器上），web服务器可以是你的pc，不过大多是ISP提供的web主机；

**ISP**

连接Internet的速度，24小时支持，备份重要数据，流量带宽，电子邮件，数据库访问，安全和稳定性。。。

磁盘空间：图片，音视频文件大小等；

月流量：访问量，超过限制了网站会关闭，额外费用，容易升级吗？

签合同之前可以访问下其他客户或网站；

图片服务器：分离负载模块；

**CDN：内容分发网络，**通过在网络各处放置 [节点服务器](http://baike.baidu.com/item/%E8%8A%82%E7%82%B9%E6%9C%8D%E5%8A%A1%E5%99%A8) 所构成的在现有的互联网基础之上的一层智能[虚拟网络](http://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E7%BD%91%E7%BB%9C)，CDN系统能够实时地根据[网络流量](http://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%B5%81%E9%87%8F)和各节点的连接、负载状况以及到用户的距离和响应时间等综合信息将用户的请求重新导向离用户最近的服务节点上。

**域名**

去域名注册公司买一个；

**在本地部署网站**

控制面板，打开或关闭windows功能，IIS服务；

打开IIS管理器，右键网站，将html/css/js文件路径添加进去，ip/port配置一下，局域网内即可访问了；

### 标签

#### 概念

标签的含义，分类，属性，事件，上w3c 查手册；

全局属性，每个标签都可以指定的，如id（标签唯一）,class,style（css相关）;

相似标签的区别，如<span>,<p>,<div>；

不同浏览器支持的h5标签不同：支持度不同，名字不同。。。；

#### 常用标签一览

##### 头部和元信息

<**script**脚本><**style**内部样式><**link**外部样式><**title**标题><**base**链接前缀target,href><**meta**元数据>

元数据属性

**http-equiv** 元数据名称（浏览器专用）

<meta **http-equiv=**”**Expires**” content=”Sat Sep 27 16:12:36 CST 2008”/> 网页过期时间

**http-equiv=**”**Pragma**” content=”no-cache” 禁止缓存网页

**http-equiv=**”**Refresh**” content=”2” 2s后自动刷新网页

**http-equiv=**”**Set-Cookie**” content=”name=value expires=过期时间, path=/” 设置Cookie，过期Cookie会删除

**http-equiv=**”**content-Type**” content=”text/html;charset=GBK” 页面内容和字符集

**http-equiv=”X-UA-Compatible”** content=””强制浏览器按照特定标准渲染网页；

name=”generator页面生成工具/description站点内容/keywords搜索引擎关键词”

content 值

##### body标签

<**style**样式><**h1-6**标题><**p**段落><**br**换行><**hr**水平线><**div**块><**span**块>

<**b**粗体><**i**斜体><**em**强调><**strong**粗体（增强语义）><**small**小字><**sup**上标><**sub**下标><**bdo**文本方向><**mark**重点><**time**时间，如果内容不是yyyy-mm-ddThh:mm:ss格式，就要datetime属性>

<**abbr**缩写><**address**地址><**blockquote**长引用><**q**短引用><**cite**作品标题><**code**代码><**dfn**专用术语><**del**删除的文本><**ins**插入的文本><**pre**保留格式><**samp**示范文本><**kbd**键盘输入的文本><**var**变量><**article**文章<**header**,**footer**,**section**,**article**子元素互相嵌套>><**nav**导航条><**aside**附属信息><**figure**图片区域><**figcaption**标题>

<**a**超链接**href** 资源url,target在哪里打开,media资源类型> href=”xx.html#锚点(id)”

<**ul**无序列表><**ol**有序列表><**li**列表子元素><**dl**普通列表><**dt**列表标题><**dd**列表项>

<**img**图像 src图片url,alt图片文本,height,width><**map**图片映射><**area**图片映射区域>

<**table**表格 cellpadding内容与边框间距,cellspacing单元格间距,width><**caption**标题><**thead**头><**tfoot**><**tr**一行><**tbody**表格体><**td**单元格 colspan横跨多少列,rowspan><**th**页眉(td加粗)><**colgroup**整体指定属性><**col**子元素>

<**iframe**内联框架显示资源 src>

<**meter**计数仪表图形><**progress**进度条>

**示例：**

<div><bdo dir=”ltr”><b onclick=..>文本</b></bdo><br/></div> keyword：标签、属性、事件、内容、嵌套；

<img src=.. usemap=”#map名”> keywrod：配合

<map name=...>

<area ..>

#### 常用属性一览

**id**唯一标识，**style**样式，**contentEditable**=”true”可编辑（继承），**designMode**=’on’整个html可编辑，**hidden**=”true”，**draggable**=”true”可拖拽

#### h5拖拽api

##### 简要

拖动能携带数据，交互更强大，指定draggable=”true”即可，然后为拖动元素绑定事件监听器；

##### 使用js控制拖放

事件一览

**ondragstart**开始拖动；**ondrag**在拖动；**ondragend**结束；**ondragenter**拖动时鼠标经过的元素触发，进入；**ondragover**在范围内；**ondragleave**离开；**ondrop**范围内结束；

<div id="abc" draggable="true">abc</div>

<script type="text/javascript">

var btn = document.getElementById("abc");

btn.ondragstart = function(evt){//绑定事件

evt.dataTransfer.setData("text/plain","abc");

}

document.ondragover = function(evt){

return false;//取消事件默认行为

}

</script>

拖放事件有一个**dataTransfer**属性，它是一个DataTransfer对象，属性和方法有：

**dropEffect**拖放目标上的拖放行为：none、copy、link、move

**effectAllowed**拖放元素的行为：none、copy、copyLink、copyMove、link、linkMove、move、all、uninitialized；

**items**拖放数据DataTransferItems对象；

**setDragImage**设置拖放图标；

**addElement**添加图标；

**types**返回DOMStringList对象，包含了所有数据类型；

**getData**获取数据；

**setData**设置数据；

**clearData**清除数据；

#### h5多媒体

##### 简要

以前网页播放音视频通常借助于第三方插件，如flash，或者开发者自己开发多媒体播放插件；

现在h5提供audio和video，各浏览器厂商实现了内置的播放器，但是音视频格式 涉及相应厂商的专利，浏览器无法自由使用它们的解码器；

<audio controls> //兼容写法

<source src="筷子兄弟 - 我一定要得到你.mp3" type="audio/ogg">

</audio><br/>

<video src="想变得可爱.mp4" controls>不支持</video><br/>

<**audio/video**音频 src资源URL，autoplay自动播放，controls播放控制条，loop循环，preload预加载=auto全部，metadata元数据，指定autoplay忽略，poster视频用，播放之前的海报，width视频用，height>

##### 使用js控制播放

audio元素对应HTMLAudioElement对象，video-HTMLVideoElement，支持的方法：play，pause，load，canPlayType浏览器是否支持类型播放；

**属性一览**

**buffered**返回TimeRanges对象，浏览器缓存的数据；**currentSrc**播放资源的URL；**currentTime**时间点秒；**defaultPlaybackRate**播放速度；**duration**持续时间秒；**ended**播放结束否；**error**有错误返回MediaError对象；**muted**静音否；**networkState**下载资源状态；**paused**暂停否；**playbackRate**播放速度；**played**返回TimeRanges对象，已播放时间段；**readyState**准备状态；**seekable**返回TimeRanges对象；**seeking**是否在定位；**startTime**开始时间；**volume**音量；

注：**TimeRanges**是时间段数组，length，start(index)，end(index)第index个时间段的开始结束时间；

**事件一览**

**onabort**播放器下载数据中止了；**oncanplay**播放中间缓冲数据时；**oncanplaythrough**不缓冲；**ondurationchange**文件长度改变；**onemptied**元素为空（网络、加载错误）；**onended**播放结束；**onerror**出错；**onloadeddata**加载完成后；**onloadedmetadata**元数据加载完成后；**onloadstart**开始加载；**onpause**暂停；**onplay**播放；**onplaying**正在播放；**onprogress**正在加载；**onratechange**速度变化；**onreadystatechange**状态改变；**onseeked**成功定位后；**onseeking**在定位；**onstalled**延迟；**onsuspend**未获取全部数据停止；**ontimeupdate**播放位置改变；**onvolumechange**音量改变；**onwaiting**暂时得不到下一帧；

### 表单

#### 简要

收集用户输入，作为请求参数提交到服务器；

#### 表单元素

<**form**表单action提交的地址,method请求类型,enctype字符集,name表单名称,target>

注：

1），请求类型 GET,POST，GET会把请求参数以name-value字符串附加到原URL后，POST则放到html header中；

2），enctype类型，application/x-www-form-urlencoded 默认，只处理控件的value值，multipart/form-data 以二进制流处理，文件域的文件内容也作为请求参数，text/plain，当action是<mailto:URL>时；

3），每个有name属性的控件对应一个参数，相同name一个参数多个值；

如：<http://www.xxx.org/?username=abc&username=&ok=%CC%E1%BD%BB>，?开始，username是相同name，&隔开，value为空不填，%CC..是中文，用了默认编码；

<**input**输入控件 type类型，checked，disabled，maxlength，readonly，size宽度，src>

注：

1)，type类型，单行文本**text**，密码输入**password**，隐藏域**hidden**（提交额外请求参数），单选**radio**，复选**checkbox**，图像域**image**（有图像的提交），文件域**file**，提交、重设、无动作**submit**，**reset**，**button**，颜色选择器**color**，日期选择器**date**，时间选择器**time**，UTC日期时间**datetime**，**datetime-local**，第几周**week**，**month**，**email**（multiple属性），**url**，**number**，拖动条**range**（min，max，step）；

<**label**标签>有语义，点击获得焦点；

<label for="username">用户名：</label>

<input id="username" name="username" type="text" /><br/>

<label>密码框：

<input id="pswd" name="pswd" type="password" />

</label><br/>

<**button**按钮 type类型，name，value，disabled>，可包含文本格式化标签，图像等，不要放图像映射；

<**select**列表框或下拉菜单multiple，size显示多少项（指定任一个就显示列表框）><option列表项 disabled，selected，value><optgroup项组 label，disabled>

<**textarea**文本域>，文本内容作为value值；

<**input type=”file”**文件上传域 accept 文件MIME类型，multiple>，多个文件，js可访问文件内容，P69；

#### 表单属性

**form**=”form name”，控件属于哪个表单；formaction提交给不同的action；**formxxx**属性：动态改变，formenctype ，formmethod，formtarget；

**autofocus**自动获得焦点；**placeholder**默认提示信息；

**list**属性，一个ComboBox功能，值=<datalist>的id，<datalist>和select一样，只是它看不见；

**required**必填，**pattern**符合正则，**min**、**max**、**step**数值、日期类型范围；

**name/id属性不能和对象的原有方法和属性名相同，会导致失效；**

#### 表单校验

1），required，pattern，min、max、step等属性；

2），checkValidity()表单控件方法，P76；

### 绘图

#### 简要

h5新增<canvas画布>，获取CanvasRenderingContext2D对象，使用js控制绘图；

<canvas id="mycanvas" width="300" height="180" style="border:1px solid black">

<script type="text/javascript">

//三步：1）获取Canvas对象，2）获取CanvasRenderingContext2D对象，3）调用api绘图

var mc = document.getElementById('mycanvas');

var ctx = mc.getContext('2d');

ctx.fillStyle = '#f00';

ctx.fillRect(30,40,80,100);

</script>

</canvas>

#### 绘图api一览

<https://developer.mozilla.org/zh-CN/docs/Web/API/CanvasRenderingContext2D>

**arc**画弧；**arcTo**画弧；**beginPath**定义路径；**closePath**结束路径；**bezierCurveTo**贝塞尔曲线；**clearRect**擦除矩形区域上的图形；**clip**裁剪画布；**createLinearGradient**线性渐变；**createPattern**平铺；**createRadialGradient**圆形渐变；**drawImage**绘制位图；**fill**填充路径；**fillRect**填充矩形；**fillText**填充字符串；**lineTo**画线；**moveTo**移动点；**quadraticCurveTo**二次曲线；**rect**画矩形；**stroke**画路径边框；**strokeRect**画矩形边框；**strokeText**画字符串边框；**save**保存绘图状态；**restore**恢复绘图状态；**rotate**旋转坐标系；**scale**缩放坐标系；**translate**平移坐标系；

#### 绘图属性一览

**fillStyle**填充样式；**strokeStyle**画笔样式；**font**字体；**globalAlpha**透明度；**globalCompositeOperation**叠加效果；**lineCap**线条端点样式；**lineJoin**线条拐点样式；**miterLimit**拐点样式为meter控制；**lineWidth**线宽；**shadowBlur**阴影模糊度；**shadowColor**阴影颜色；**shadowOffsetX**阴影偏移；**textAlign**文本水平对齐方式；**textBaseAlign**垂直对齐方式；

## [CSS](#目录)

### 简要

css3.0是最新标准，控制html文档显示的；

4种用法：

1. **链接外部**（css与html分离）；

<link rel="stylesheet" href="test.css" type="text/css" />

1. **导入外部**；

<style type="text/css">

@import url("mycss.css");

</style>

1. **使用内部**（head下定义style）;
2. **使用内联**（控制部分元素）；

<div style="font-size: 18px;

color: #60c;

height: 30px;">

内联语法格式：style=”属性1：值1;属性2：值2...”

定义样式的格式大致如下：

selector { //对哪些元素作用

property1 : value1;

property2: value2;

...}

所以css就是掌握selector定义方法 和 css样式；

效果是叠加显示的，取优先级高的，各浏览器支持度不同；

**单位**

**像素**：1英寸=90像素，和分辨率有关；

**em**：本元素字体的font-size，如元素font-size=14，那1em=14px；如果font-size本身以em为单位，则取父节点的；

**%**：font-size \* %；

### 选择器定义

**E{}**：元素；

**E[attr]{}**：对具有attr属性的元素；

**E[attr=value]{}**：对attr=value的元素；

**E[attr ~=value]{}**：attr是系列值，以空格隔开，对attr的某个值=value的元素；

**E[attr |=value]{}**：attr是系列值，以|隔开，对attr第一个值=value的元素；

**E[attr ^=”value”]{}**：对attr的值是以value字符串开头的元素；

**E[attr $=”value”]{}**：value字符串结尾；

**E[attr \*=”value”]{}**：包含value字符串；

**#id{}**：选择id；

**E#id{}**：元素内id；

**.class{}**：选择class；

**[E].class{}**：元素内class；

**sel1 sel2 {}**：选择器内的选择器；

**sel1>sel2{}**：选择器内的子元素；

**sel1~sel2{}**：选择器的兄弟元素；

**sel1,sel2..{}**：选择器组合；

**sel1:not(sel2)**：符合sel1但不符合sel2；

**sel:root/first-child/last-child/nth-child(n)/nth-last-child(n)/only-child/first-of-type/last-of-type/nth-of-type(n)/nth-last-of-type(n)/only-of-type/empty**：child是子元素，type是兄弟元素，nth还可以指定n=odd/event奇偶，3n+1(n=0,1,2…)；

**sel:link/visited**：未访问过的/访问过的超链接；

**sel:active/hover**：鼠标点击未释放/悬停/获得焦点的元素；

**sel:focus/enabled/disabled/checked/default/read-only/read-write/selection**：获得焦点/可用/不可用/选中/默认选中/只读/读写/选中内容；

**sel:target**：锚点目标元素；

**:first-letter**：对象的第一个字符；

**:first-line**：第一行，和first-letter一样，只对div,p,section等起作用，span需要指定宽高或position或display；

**:before**：对象前端插入数据；

**:after**：尾端；

### css属性一览

#### 其它

**content**元素之前或之后插入内容：字符串、url(url)、attr(alt)、counter(name)、counter(name,list-style-type)、open-quote、close-quote等；**quotes**为content定义open/close-quote的，两个空格隔开的字符串；**counter-increment**计数器；**counter-reset**计数器复位；

div>div{ counter-increment: cnt; } //quotes,counter-increment,counter-reset是配合content使用的；

div>div:before{ content:counter(cnt) ') '; / content:'第' counter(cnt,upper-roman) '个 '; }

counter-reset属性用在多级编号，在父元素内重置；

#### 文字

**font**字体 font-style font-variant font-weight font-size line-height font-family；**color**颜色；**font-family**字体（浏览器内嵌，逗号隔开的多个字体，优先）；**font-size**大小，xx-small/large/../%父组件百分比/11pt/px；**font-size-adjust**字体大小微调；**font-style**风格，normal/italic斜体/oblique倾斜；**font-weight**加粗，lighter/normal/bold/bolder/100；**font-stretch**字体横向拉伸，narrower压缩/wider拉伸；**text-decoration**修饰线，none/blink闪烁/underline下划线/line-through中划线/overline上划线；**font-variant**大写字母格式，normal/small-caps；**text-shadow**文字阴影；**text-transform**大小写，none/capitalize首字母大写/uppercase全大写/lowercase全小写；**line-height**行高；**letter-spacing**字符间距；**word-spacing**单词间距；

**text-shadow**:color xoffset yoofset length / x y radius color；多个阴影逗号隔开；

**font-size-adjust：**同样16px大小，宽度不同，小写x高度/16px，固定；

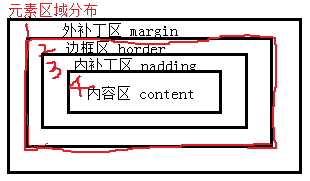
**color：**，#88/red/rgb()，hsl hue色调0红色120绿色240蓝色，saturation饱和度%，lightness亮度%，rgba，alpha透明度0~1，hsla；

**text-indent**缩进；**text-overflow**文本溢出，clip裁剪/ellipsis省略号；**vertical-align**垂直对齐，auto/baseline/sub/super/top/middle/bottom/length；**text-align**水平对齐，left/right/center/justify两端对齐；**direction**文本方向，ltr/rtl；**word-break**换行方式，normal浏览器规则/keep-all/break-all单词间换行；**white-space**空格处理，normal/nowrap不换行；**word-wrap**内容换行，normal/break-word；

**@font-face**{ font-family:name任意; src: local(url), ur(url) format(“TrueType/OpenType”);}服务器字体，优先使用本地；

#### 背景

**background**背景；**background-attachment**背景图片滚动否，scroll/fixed；**background-color**颜色；**background-image**图片；**background-position**图片位置；**background-repeat**平铺，repeat/no-repeat/repeat-x/repeat-y；**background-clip**背景覆盖范围；**background-origin**背景覆盖起点；**background-size**背景大小，20px/80%父组件/auto纵横比；



**background-clip/background-origin**，border-box边框以内/no-clip边框以内/padding-box内补丁以内/content-box内容区；

可以指定多张背景图片，及它们的重复，位置等属性；

#### 边框

**边框**

**border**粗细线型颜色复合；**border-color**；**border-style**线型，none/hidden/dotted点线/dashed虚线/solid实线/double双线/groove3D凹槽/ridge3D凸槽/inset3D凹入/outset3D凸入；**border-width**；**border-top/right/bottom/left**上右下左边框复合；**border-top-color/border-top-style/border-top-width**；**border-radius**圆角边框；**border-top-left-radius**；

**padding/margin**内/外补丁边距；**padding/margin-top**；

粗细线型颜色 指定4个参数，上右下左，2个参数，上下左右，3个参数，上左右下，1个参数，所有；

圆角指定2个参数，左上右下，3个参数，左上右上左下；

**height**；**max\_height**；**min\_height**；**width**；内容区宽高；**box-sizing**指定区域，content-box/padding-box/border-box；**resize**可拖动改变大小否，组件要设置overflow属性，none/both/horizontal/vertical/inherit继承父组件默认；

#### 布局&盒模型

**盒模型**

能呈现一片空白的元素，都可以当盒模型，盒子模型和布局不冲突，div，span，section等，

**盒子的3D模型**

从上到下依次：border，content+padding，bg-image，bg-color，margin；

display属性可控制其外观，none/block/inline/inline-block/inline-table；

**block**，默认占据一行；

**inline**，一行可以有多个组件；

**none**隐藏对象，页面空间释放；

**inline-block**既不一行又能指定宽高，默认底端对齐，vertical-align:top；

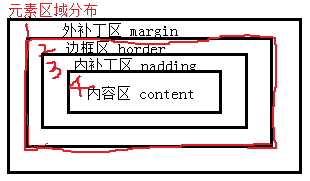
**inline-table/table/table-caption/table-cell/table-column/table-column-group/table-header-group/table-footer-group/table-row/table-row-group**，表格既不一行又能指定宽高，将组件显示为表格元素；

**list-item**将组件转化为列表元素，list-style-type序号类型；

**box-shadow**盒阴影，hOffset水平偏移/vOffset/blurLength模糊度/scaleLength缩放度/color；

**columns**分栏复合宽和数，**column-width/column-count**，**column-rule**分隔条复合宽样式颜色，**column-rule-width/style/color**，**column-gap**栏间距，**column-fill**栏高，auto/balance统一成内容最多的；

**布局**



有标准的文档流、float属性，position属性等；

**标准文档流**

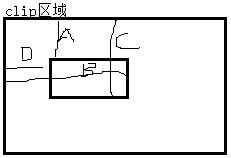
标签元素（div，ul，li，dl，span...），触碰到边缘会自动换行，一个元素就是一个“盒子”，有margin，padding，border，content；

**浮动布局**

**float**，会左右移动，直到碰到容器（父元素）边缘，仍然是标准文档流中，**left**/**right**属性；

设置了float，元素的宽度随内容变化，**受其影响的只有它后面紧邻的元素**；

**clear**清除浮动（对紧邻的后面的元素），none/left/right/both都不允许；**clip**裁剪，auto不裁剪/rect(A,B,C,D)该区域显示，**overflow**要设置hidden；**overflow/overflow-x/y**组件不够容纳内容时显示方式，visible不剪切不加滚动条/auto加滚动条/hidden裁剪/scroll总是显示滚动条；**visibility**元素显示隐藏，元素页面空间保留；



**绝对布局**

**position**（**层模型**），absolute绝对定位/relative相对前一个对象（仍在文档流中）/static（默认）；

absolute不在文档流中了，宽度随内容变化；

偏移：无已定位的祖先元素，以<html>为偏移参照基准，有已定位，则参照最近的定位祖先元素偏移；

**z-index**漂浮层数；**top/left/right/bottom**偏移；

**outline**轮廓颜色线型线宽；**outline-color/outline-style/outline-width/outline-offset**轮廓与边框的距离；

#### 表格&列表&光标

**border-collapse**边框合并还是分隔，separate/collapse；**border-spacing**边框间距；**caption-side**标题位置top/bottom/left/right；**empty-cells**空内容是否显示边框separate时有效，show/hide；**table-layout**宽度布局，auto/fixed；

**list-style**复合序号图片位置类型；**list-style-image/position/type**，outside/inside；

改变光标在组件上的形状；

**all-scroll**十字箭头/**col-resize**水平拖动线/**crosshair**十字线/**move**移动十字线/**help**问号箭头/**no-drop**禁止光标/**not-allowed**禁止/**pointer**手型/**progress**沙漏/**row-resize**垂直拖动/**text**文本编辑/**vertical-text**旋转90度/**\*-resize**各种方向拖动光标，w/s/n/e/ne/sw/se/nw，上北下南；

#### media query

对方设备、浏览器等原因，如分辨率，色深，最终效果非常丑陋，css定义了各种媒体类型，针对不同设备编写；

语法：

**@media [not|only] 设备类型 [and 设备特性] \*（出现0~N次）**

**设备类型**：all/aural语音和音频合成器/braille触觉反馈/embossed盲文印刷/handheld手提/print打印机/projection投影图像/screen计算机显示器/tty固定间距字符设备/tv电视类；

**设备特性：**width/min/max-width浏览器宽/height/aspect-ratio宽高比/device-width设备分辨率/device-height/device-aspect-astio/color设备色深（多少位）/color-index色彩表颜色数/monochrome单色帧缓冲器中每像素位数/resolution分辨率值/scan扫描方式，progressive逐行/interlace隔行/grid基于栅格否；

#### 变形与动画

**transform**设置变形，**translate**(tx[,ty])平移/**translateX**/Y/**scale**(sx,sy)缩放比/scaleX/Y/rotate(angle,30deg)顺时针旋转/**skew**(sx[,sy])横/纵向倾斜/skewX/Y/**matrix**(m11,m12,m21,m22,dx,dy)变换矩阵/**transform-origin**变形中心点，xCenter/yCenter=left/top/right/bottom/center/长度值/%/left top组合；

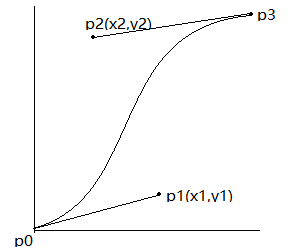
**矩阵变换公式**：(x,y) \*( ) = (x\*m11 + y\*m21, x\*m12 + y\*m22); 再平移dx,dy；

**transition**设置动画，**transition-property**对哪个属性进行平滑渐变，width,bg-color等/**transition-duration**/**transition-timing-function**渐变速度，ease慢-快-最快-慢/linear匀速/ease-in慢-快/ease-out快-慢/ease-in-out同ease/**cubic-bezier**(x1,y1,x2,y2)贝塞尔曲线控制；

**同时指定多个，不同属性用分号，同属性指定多个用逗号；**

transition: left 5s linear, top 5s linear;

transform:rotate(24deg);transform:translate(5px);



贝塞尔曲线，x轴：时间，y轴：属性完成度；

Animation动画，animation-name已有关键帧/animation-duration/animation-timing-function速度/animation-delay多久开始执行/animation-iteration-count循环次数；

animation-name格式：

@keyframes 帧名称{

from | to | % {//关键帧位置

属性n：值n;}}

## [JavaScript](#目录)

### 概念

桌面，游戏机，平板，手机。。。它们的Web浏览器都有JavaScript解释器，它是一种脚本语言，浏览器动态解释执行（无须编译）。

运行方式：

1，<a href=”javascript:代码”>...

2，<script type=”text/javascript”>代码</script>

3，<script src=”xx.js” type=”text/javascript”></script>

代码有错误，浏览器会指出来的；

### 语法

#### 注释&变量&数据类型

**注释**

// /\*\*/

**变量**

var a,b=0,c;

如果不使用var，程序会先查找上下文是否存在变量，不存在则定义一个；

**数据类型**

**数值**

5e2, 0x13, 014, Infinity正无穷（数值超过范围）;

与运算得NaN，各自相等，两个NaN不相等，0/0得NaN，/0得；

Number.MAX\_VALUE, POSITIVE\_INFINITY, NAN...；

浮点数用差值比较（小于一个值即可认为相等）；

**字符串**

单/双引号一样，相邻使用了就一个单一个双；

JS以String类包装方法属性有：**String**()构造/**charAt**()指定索引字符/**charCodeAt**()指定索引字符的Unicode字符/**length**/**toUpperCase**()/**toLowerCase**()/**fromCharCode**()静态方法，Unicode转字符串/**indexOf**()字符串第一次出现位置/**lastIndexOf**()/**substring**()/**slice**()返回子串/**match**()正则匹配子串/**search**()正则匹配子串/**concat**()拼接/**split**()分隔/**replace**()；

s.slice(0,-1)取第一个到最后一个字符的串；

s.charCodeAt(n)取第n个字符的Unicode值，汉字是1个长度；

a = s.match(/[a-z][+@d.m/g);search](mailto:+@d.m/g);search) 返回索引/-1，match返回子字符串/数组（使用g）/null；

**布尔值**

true，false；

**undefined&null**

undefined表示不存在或未分配值，null是存在，只是值为空；

undefined == null 返回true，undefined === null 返回false；

**对象**

内置对象有：Array数组/Data日期/Error错误/Function函数/Math数学/Number数值/Object对象/String字符串；

**数组**

var a=[3,3.14,”str”]; var b = new Array(); c[100]=’a’;

不会越界，未赋值元素返回undefined，类型可以不同；

**函数**

可以当一个函数/类/对象，代码复用的基础；

#### 类型转换

隐式：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 值 | 目标类型 | | | |
| 字符串 | 数值 | 布尔型 | 对象 |
| undefined | “undefined” | NaN | false | Error |
| null | “null” | 0 | false | Error |
| 字符串 | 不变 | 数值或NaN | true | String |
| 空字符串 | 不变 | 0 | false | String |
| 0 | “0” | 0 | false | Number |
| NaN | “NaN” | NaN | false | Number |
| Infinity | “Infinity” | Infinity | true | Number |
| -Infinity | “-Infinity” | - Infinity | true | Number |
| 数值 | 数值字符串 | 不变 | true | Number |
| true | “true” | 1 | 不变 | Boolean |
| false | “false” | 0 | 不变 | Boolean |
| 对象 | toString()返回值 | valueOf(),toString()或NaN | true | 不变 |

隐式虽然方便，但可读性非常差，比如var a = “3.14”; var b = a – 2;//a转成数字 var c=a+2//2转成字符串；

显示：

toStirng()，parseInt()，parseFloat()；

#### 运算符

大致同c++；

**赋值**：**=**；

**算术**：+ - \* / % Math对象方法，++--，+&>>=。

**位运算**：&|~<< >> >>>无符号；

**比较**：!=值不等，!==类型和值都不等，===严格相等；

**逻辑**：&&||!；

三目：?:；

**逗号**：表达式返回最右边的值；

**void**：强行指定表达式不会返回值，如：a = void(b=5,c=7); a=undefined;

**typeof/instanceof**：typeof(a)或typeof a，返回数据类型，

|  |  |
| --- | --- |
| 数据值 | 返回类型 |
| undefined | undefined |
| null | object |
| 布尔值 | boolean |
| 数字 | number |
| 字符串 | string |
| 对象 | object |
| 函数 | function |

instanceof变量是否为某个类的实例，如：a instanceof Array;

#### 语句

**语句块**：{}，不能作为变量的作用域的；

**异常**：js异常都是Error对象，使用throw new Error(errString);抛出异常，机制和c++类似，寻找catch，找不到就给浏览器，程序终止；

try{} catch(e){} finally{}

try后只能有一个catch，e.message就是errString；

**with**：避免重复书写对象， with(object){ }，如：

with(document){ wirteln(); getElementById(); } //省略了document.

**流程控制**：大致同c++，if-else，switch，do-while，for，break，continue，

**for in**遍历元素或者对象的属性，for(索引/属性 in object) {}

**break/continue 标识符**; 跳转到标识处，如

outer:

for(...){ for(...) { break outer; }}

#### 函数

##### 定义&调用&参数

**定义**

1）function 名(参数列表){ } //同时定义了函数、对象、方法、类

2）var f = function(参数列表){}; //匿名函数，只定义一个函数对象

可以定义局部函数（函数内定义）；

**调用**

直接调用：obj.method();

**call调用**

var each = function(arr, fn) {

var i = ..;

fn.call(null, i); //函数.call(调用者，参数1，...);null是window对象

}

函数.call(调用者，参数1，...) = 调用者.函数(参数1，...)

**apply调用**

和call大致相同；

var f = function(a,b){}

var g = function(m,n){ f.apply(this, arguments); } //用arguments代表传入的所有参数，其实它是一个数组，所以f.apply(window, [12,23]); 也是对的；

**参数**

按值传递，只是复合类型它本身就是一个引用；

参数在调用的时候可以不传，它的值是undefined；

由于js是弱类型，因此编写健壮代码有时候要这样：if(typeof xxx == “object” && typeof xxx.nn == “number”){}

##### 函数、对象、方法、类

**定义一个函数，同时定义了函数、对象、方法、类；**

var test = function(name){ return “你好,”+name; }

var rval = test(“myn”); //函数

var obj = new test(“myn”); //同名的类构造

function Person(name, age){

this.name = name; //this，**对象成员**

Person.age = age; //类名，**static成员**，用Person.age访问

this.info = function(){ ... } //**成员方法**，方法附加到对象，默认附加到window对象上

}

var p = new Person(“myn”,26);

p.info();

p.weight = 120; //**动态增加属性**

**对象的本质是一个map**，key是成员属性名/成员函数名，可以用**obj[name]**访问；

**函数是一等公民，就算它定义为成员方法，也不会从属于其他类、对象；**

function Dog(name,age,bark){

this.name = name; this.age = age; this.bark = bark;

this.info = function() { return this.name; }

}

var dog = new Dog(“旺财”,3,”汪汪”);

function Cat(name,age){

this.name = name ; this.age = age;

}

var cat = new Cat(“甜心”,2);

**alert(dog.info.call(cat));** //dog.info翻译为Dog类的info函数，call(cat)翻译为this变成了cat;

**没有继承；**

都是Object类的子类，可以动态的添加属性和方法；

function Person(name, age){

this.info = function(){ } //这种定义方式每次创建对象时，都会创建一个info函数，导致系统有多个info函数，性能低下；

}

**模仿继承**：为类的prototype属性添加函数、属性；

Person.prototype.Fun1 = function(p1){ }

扩展后才起作用，之前的对象没用；

**JSON创建对象/数组**

**var object = {**

**属性n : 值n,**

**...**

**};**

**var arr = [值n,...];**

如：

person = {

name:”myn”,

son : { //看到{}是对象

name:”hh”, //看到:是属性值

grade:1

},

info:function(p1){ ... }

}

### DOM

#### 概念

文档模型，提供访问、修改文档的编程接口，浏览器厂商都内置DOM解析器：将磁盘上的文档（xml/html等）转换成内存中的DOM树，当浏览器加载html时，内存中已经有DOM树了；

#### js提供的接口

##### 简要

js用对象封装标签，Node、Document、Element、HTMLElement四个父类，其他子类：HTMLDocument整个文档/HTMLBodyElement<body>控件/HTMLDivElement/Form/Select/Option/IFrame/InputFrame/Table.....；

##### 访问

###### 简要

1）**document.getElementById(id);** //返回HTMLXxx对象

2）Node **parentNode**父节点/**previousSibling**上一个兄弟节点/**nextSibling**/Node[] **childNodes**所有子节点/Node[] **getElementsByTagName**(tagName)具有指定标签名的所有子节点/**firstChild**/**lastChild**

**换行、空白等，也会被HTML当作节点！**

###### 表单&列表框&表格

**HTMLFormElement**，额外属性有：**action**/**elements**全部表单控件的数组/**length**=elements.length/**method**/**target**；额外方法：**reset**()/**submit**()；

elements是一个**HTMLCollection**对象，它既可以当数组（索引）也可以当map（key值），如elements[“pass”]控件name或id值=”pass”的，如果有多个，继续返回elements，很多返回对象数组的地方其实都是它；

**访问表单元素**：elements[i]，elements[“name/id”]/，HTMLFormElement.name/id；

**HTMLSelectElement**，额外属性：**form**表单对象/**length**/**options**选项数组/**selectedIndex**选中索引/**type**类型select-multiple或select-one；

**访问列表框元素**：select.options[i]，**HTMLOptionElement**，属性有：**form/defaultSelected**是否默认选中/**index**/**selected**是否被选中/**text**/**value**；

**HTMLTableElement**，额外属性：**caption/rows/tBodies/tFoot/tHead**；

**访问表格元素**：table.rows[i]，**HTMLTableRowElement**，属性：**cells/rowIndex/sectionRowIndex**该行索引，**HTMLTableCellElement**单元格，**cellIndex**该单元格索引；

###### 增删改

**修改属性**

属性：**innerHTML**当元素开始结束标签之间只是字符串时，元素内容，但将其置为null也会删除控件所有选项/**value**表单控件内容/**className**选择器/**style**修改内联css；

**增加节点**

document.**createElement**(“div”);

Node **cloneNode**(boolean deep); //开销略小

**appendChild**(Node);

**insertBefore**(Node, Node 参照节点);

**replaceChild**(Node, Node old);

HTMLSelectElement有add方法：**add**(HTMLOptionElement, HTMLOptionElement before)，before=null追加 **或者**var op = new Option(...); a.options[i] = op；

HTMLTableElement有如下方法：

**insertRow**(index)在索引处插入一行，返回HTMLTableRowElement，**CreateCaption**/**CreateTFoot**/**createTHead**/**insertCell**(index)；

**删除节点**

removeChild(Node)删除Node子节点；

HTMLSelectElement有remove方法，或者 options[i] = null; 或者 innerHTML=null;

HTMLTableElement有deleteRow，deleteCell；

##### window&document对象

全局变量在其内，只是在window内，如果一个页面有多个frame（多个window），它们互不影响；

属性/方法：**alert**()/**confirm**()/**prompt**()弹出警告确认和提示输入对话框/**close**()关闭窗口/**focus**()/**blur**()让窗口获得失去焦点/**moveBy**()/**moveTo**()移动窗口/**open**()打开新窗口/**print**()打印窗口/**resizeBy**()/**resizeTo**()重设窗口/**scrollBy**()/**scrollTo**()滚动文档/**setInterval**()/**clearInterval**()定时器每隔/**setTimeout**()/**clearTimeout**()定时器执行一次/**closed**/**defaultStatus**/**status**状态栏的文本/**document**/**frames**[]/**history**浏览历史/**location**文档URL/**name**窗口名字/**navigator**浏览器/**parent**返回本frame的直接父窗口/**screen**屏幕对象/**self**/**top**返回包含本frame的顶级窗口；

history是**History对象**，方法：back()/foward()/go(i)前进后退多少页；

**location对象**，方法：hostname主机名/href/host/port/pathname/protocol

**screen对象**，属性：width/height/colorDepth；

**confirm()**点击确定返回true，prompt()返回字符串；

open()语法，略；

**setInterval**(“code”,ms)/clearInterval(timer);

**Navigator对象**，**appName**内核名/**appVersion**浏览器版本/**platform**所在操作系统/

**geolocation**地理对象，方法：**getCurrentPosition**(成功回调，失败回调，可选参数)/int watchCurrentPosition()持续监听地理位置，返回监听器id/clearWatch(id)；

回调函数形如：function(position){}，position对象属性：timestamp/coords是**Coordinates对象**，属性：longitude/latitude/altitude高度值/accuracy精度m/altitudeAccuracy/speed浏览器所在设备的移动速度/heading设备移动方向/timestamp；

失败回调：function(error){} ，error属性：code错误代码：用户拒绝、无法获取、超时/message错误描述；

额外选项，一个js对象，包含属性：enableHighAccuracy是否要求高精度/timeout超时时长/maximumAge地理位置的缓存时间；

**document**方法属性：**close()/open()**打开关闭document对象/**write**()/**writeln**()输出字符串/**alinkColor**/**linkColor**/**vlinkColor/bgColor/fgColor**超链接激活、没访问过、访问过、前景背景色/**all**所有子元素/**anchors**所有命名锚点/**applets**所有Applet/**cookie**/**documentElement**返回html节点/**forms**所有表单/**frames**所有frame/**images**所有图像/**lastModified**/**links**/**location**/**referrer**上一个页面url/**scripts**所有脚本/**styleSheets**所有css/**title**/**URL**；

**cookie**

name=value对，可以是服务器传给浏览器，浏览器写入磁盘，也可以是浏览器读取，传给服务器；

name可以任意指定，有几个内置的：max-age存活时间s/expires过期时间/path路径/domain域/secure安全属性；

##### 事件

###### 简要

**绑定元素**

<标签 onXXX=”js code1; js code2; …”/>

坏处：修改了html元素和html页面；

**绑定DOM对象属性**

在js中，document.form[0].onsubmit = check; //事件属性绑定函数引用；

**事件一览**

**onabort**中断/**onblur**失去焦点/**onchange**/**onclick**/**ondblclick**/**onerror**出错/**onfocus**/**onkeydown**/**onkeypress**/**onkeyup**/**onload**装载完毕/**onmousedown**/**onmousemove**/**onmouseout**/**onmouseover**/**onmouseup**/**onreset**重置表单/**onresize**/**onselect**选择文本/**onsubmit**/**onunload**；

**改变事件的默认行为**

return false; confirm();等方法，让其返回false；

元素：click，dragdrop，keydown等等；

**触发事件**

监听器 或者 代码手动触发（调用元素的submit()/click()/blur()/select()等方法）；

###### Internet Exporer事件机制

**绑定**

1，<script **for**="btn" **event**="onclick" type="text/javascript">

alert("abc"); </script> //每个元素对应一个script for；

2，attachEvent

**obj.attachEvent(“onclick”, 函数);**//event自动传入

**obj.detachEvent(“onclick”, 函数);**//删除事件处理器

**访问事件对象**

事件对象封装了详细信息，IE中的事件对象是个全局对象：**event**，**属性**：**type**类型，”click”等/**srcElement**发生事件的元素/**clientX**鼠标x坐标/**offsetX**鼠标相对事件源元素坐标/**button**哪个鼠标键/**keyCode**键盘键/**altKey/ctrlKey/shiftKey**是否按下了/**cancelBubble**阻止冒泡/**returnValue**事件处理函数的返回值/**fromElement /toElement**鼠标移出移入的元素，对mouseover/mouseout有效；

**事件冒泡**

向父节点传递，依次调用事件处理函数，表单提交/获得焦点等没有冒泡，通过event对象的**cancelBubble=true**阻止；

**转发事件**

事件在不同节点跳跃，如转发到子节点，或者根本不在DOM上的其它节点；

**event.cancelBubble=true;**

**target.fireEvent(“onclick”, event);** //将事件重定向到target上；

**事件捕获**

事件分捕获和冒泡阶段，如果事件被捕获了，那它就不会冒泡，直接触发事件处理器直到调用release或捕获中断（浏览器失去焦点弹框显示系统菜单等），onlosecapture监听！

**target.setCapture()/releaseCapture();**

###### DOM事件机制

**绑定**

**obj.addEventListener(“click/mousedown”, hander, captureFlag,true捕获false冒泡);**

**removeEventListener**

**访问事件对象**

DOM发生事件会隐式创建一个Event对象，并作为事件处理器的第一个参数传入，因此，写跨浏览器的代码，就需要多个分支；

**DOM事件接口**

Event，MouseEvent，UIEvent，MutationEvent等，继承体系，有相应的事件；

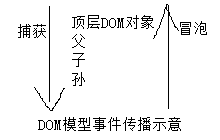
**事件属性**

**Event：type**，”click”/**target**事件源/**currentTarget**事件当前所在的事件源/**eventPhase**，Event.CAPTURING\_PHASE等/**timeStamp**发生时间/**bubbles**支持冒泡否/**cancelable**有默认行为否；

**UIEvent：view**返回window对象/**detail**数字，1单击2双击等；

**MouseEvent：button**数字，0鼠标左等/**atlKey**等/**clientX**文档坐标/**screenX**屏幕坐标/**relatedTarget**相关元素；

**事件捕获&冒泡**



**阻止传播&取消默认行为**

**event.stopPropagation();**

**event.preventDefault();**

**转发事件**

**target.dispatchEvent(Event);** Node对象方法**，**不能转发系统事件，**创建人工事件的方法**：document.**createEvent**(“Events/UIEvents/MouseEvents”);

得到事件对象后，**initEvent**(“click事件类型”,是否支持冒泡,是否有默认行为); **initUIEvent**(1,2,3,view,detail); **intiMouseEvent**(1,2,3,4,5,…);

##### 本地存储与离线应用

###### 概念

离线应用显式控制浏览器缓存哪些东西；Web Storage本地存储，离线状态下将用户提交的数据存储在本地，联网后集中提交；

cookie的不足：4KB；包含在每个http请求中，多次重复发送；

###### 接口

Web Storage分Session和Local，Session数据与用户Session（关闭浏览器或离开网页）相同，Local保存在磁盘；

**window.sessionStorage，window.localStorage属性**，它们是接口实例，属性方法：**length**多少组key-value/**key(i)**/**getItem(key)/set(key,value)/removeItem(key)/clear()**；

**存储结构化数据**

**JSON对象，JSON.parse(String)，JSON.stringify(JSObject);**

**离线应用**

像w3c school网站，离线了还能点来点去，百度搜索在断网情况下就不能用了；

### 多线程

创建一个Worker对象，浏览器即可启动一个线程；

**objWorker.postMessage(data)**给线程发送数据；

**onmessage**事件监听，收到线程数据；

**示例：**

<script type="text/javascript">

**var w = new Worker("worker.js");**

w.onmessage = function(event){

document.getElementById("result").innerHTML += event.data + ",";

};

</script>

worker.js

onmessage = function(event){ } //线程挂起，直到postMessage

**注意：**

chrome不允许**new Worker("worker.js")**这样，会报错，用来测试的js和html文件都需要发布到站点才可以正常访问；

**不能使用DOM API修改前台，可用的api有：onmessage, postMessage,importScripts(“a.js”,”b.js”)导入执行脚本，sessionStorage/localStorage，Worker嵌套，XMLHttpRequest异步请求，navigator一个WorkerNavigator对象，location一个WorkerLocation对象，self，其它与界面无关的api；**

### 网络

**跨文档消息传递**

多个html（web应用）；

targetWindow.**postMessage**(message, targetOrigin) //发送消息

**onmessage** = function(event) //接收消息

event是一个MessageEvent对象，属性：data信息/origin源域名/lastEventId发送消息的事件id/source发送消息的窗口

**WebSocket**

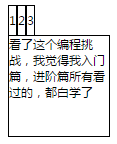
方法：**send**向服务器发送数据/**close**关闭连接；

监听器：**onopen**建立连接时/**onerror**连接出错/**onclose**/**onmessage**接收到服务器的数据；

socket状态：**readyState属性**，CONNECTING/OPEN已连接/CLOSING/CLOSED；

握手协议。。。；

## webprj练手



1,2,3用了<ul>，因为有index属性，css里修改它的盒模型能成这样（css什么不能改？），js里修改元素样式就要访问className属性，所以css里大多按class编辑样式！

## Emmet使用

1，div 按tab/ctrl+e Abbreviations补全

2，ul>li\*5 child，重复

dl>(dt+dd)\*3 括号分组

ul>li.item$$$\*5 序号001-005 ， ul>[li.item$@3\*5](mailto:li.item$@3*5) 3-7 ul>[li.item$@-3\*5](mailto:li.item$@-3*5) 7-3

3，div#header+div.page+div#footer.class1.class2.class3 id，class属性

1. 快捷键：ctrl + t（修改emmet key binding）匹配标签
2. {lorem} 实例文本

输入! + tab，h5框架